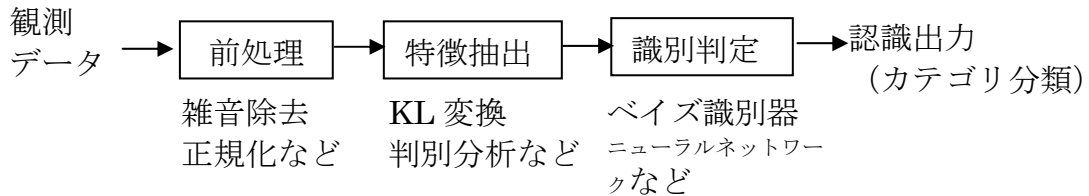


パターン認識 参考資料

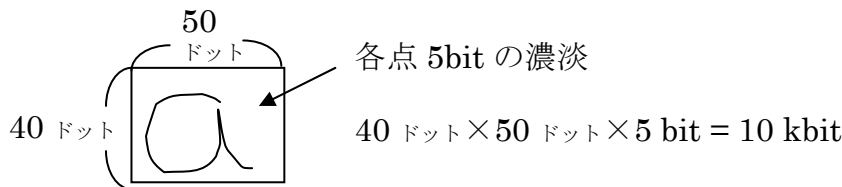
1 パターン認識とは

パターン … 文字・図形・画像・物体（3次元映像）・動画・音声・文の構造パターン

パターン認識 … 観測データから、それが予め想定されている対象のうちのいずれであるかを判定すること。



例 アルファベット文字認識



アルファベット a~z 26 種類
 A~Z 26 種類 62 種類 $\log_2 62 = 5.95\text{bit}$
 0~9 10 種類

与えられた源情報から判定に必要となる情報を保存したままでいかに情報を圧縮するかが重要

2 ベイズの識別規則

2.1 ベイズの定理（復習）

$$P(A \cap B) = P(A)P(B|A) = P(B)P(A|B)$$

$$P(B|A) = \frac{P(B)P(A|B)}{P(A)}$$

2.2 ベイズの識別規則

入力（観測）パターン … $\bar{X}(x_1, x_2, \dots, x_r)$ r 個の観測データ

入力パターンを、クラス 1 (ω_1) あるいはクラス 2 (ω_2) のいずれに属するかを判定したい。

$P(\omega_i | \bar{X})$ … 観測パターン \bar{X} が与えられたとき、そのパターンがクラス $i(\omega_i)$ に属する確率

$$P(\omega_i | \bar{X}) = \frac{P(\omega_i)P(\bar{X} | \omega_i)}{P(\bar{X})} \leftarrow \text{この確率が最大となる } \omega_i \text{ を認識結果とすればよい。}$$

ここで、分母 $P(\bar{X})$ は、クラスによらず一定なので、分子 $P(\omega_i)P(\bar{X} | \omega_i)$ が最大となる

ω_i を認識結果とすればよい。

すなわち、

$P(\omega_1)P(\vec{X}|\omega_1) \geq P(\omega_2)P(\vec{X}|\omega_2)$ ならば $\vec{X} \in \omega_1$ (\vec{X} は ω_1 に属する) と判定。

$P(\omega_1)P(\vec{X}|\omega_1) < P(\omega_2)P(\vec{X}|\omega_2)$ ならば $\vec{X} \in \omega_2$ (\vec{X} は ω_2 に属する) と判定。

言い換えると、ベイズの識別規則は以下のようなになる。

$$\begin{cases} f(\vec{X}) = P(\omega_1)P(\vec{X}|\omega_1) - P(\omega_2)P(\vec{X}|\omega_2) \geq 0 \rightarrow \vec{X} \in \omega_1 \\ f(\vec{X}) = P(\omega_1)P(\vec{X}|\omega_1) - P(\omega_2)P(\vec{X}|\omega_2) < 0 \rightarrow \vec{X} \in \omega_2 \end{cases}$$

例 下表のような統計情報が得られている。

(100人中)

X_1 : 熱	なし ($x_1=0$)		あり ($x_1=1$)	
X_2 : せき	出ない ($x_2=0$)	出る ($x_2=1$)	出ない ($x_2=0$)	出る ($x_2=1$)
ω_1 : 健康	50人	10人	5人	5人
ω_2 : 病気	2人	3人	20人	5人

$\vec{X} = (X_1, X_2) = (x_1=0, x_2=1)$ が与えられているとき、 ω_1 (健康) か ω_2 (病気) かをベイズの識別規則を用いて、判定せよ。

$$\begin{aligned} f(\vec{X}) &= P(\omega_1)P(\vec{X}|\omega_1) - P(\omega_2)P(\vec{X}|\omega_2) \\ &= \frac{70}{100} \frac{10}{70} - \frac{30}{100} \frac{3}{30} = \frac{10}{100} - \frac{3}{100} = \frac{7}{100} \geq 0 \rightarrow \vec{X} \in \omega_1 \text{ (健康)} \end{aligned}$$

2.3 離散分布の識別関数

観測パターン $\vec{X} = (X_1, X_2, \dots, X_r)'$ において、値 0, 1 をとる X_1, X_2, \dots, X_r が統計的に独立であるとき、ベイズの識別規則は線形識別関数 (X_1, X_2, \dots, X_r の 1 次関数) になる。

【証明】ベイズの識別規則では、

$$P(\omega_1)P(\vec{X}|\omega_1) \geq P(\omega_2)P(\vec{X}|\omega_2) \rightarrow \vec{X} \in \omega_1$$

と判定する。この不等式の両辺を対数化する (log をとる) と、

$$\log P(\omega_1) + \log P(\vec{X}|\omega_1) \geq \log P(\omega_2) + \log P(\vec{X}|\omega_2)$$

$$\log P(\omega_1) - \log P(\omega_2) + \log P(\vec{X}|\omega_1) - \log P(\vec{X}|\omega_2) \geq 0$$

$$\log \frac{P(\vec{X}|\omega_1)}{P(\vec{X}|\omega_2)} + \log \frac{P(\omega_1)}{P(\omega_2)} \geq 0$$

となる。よって、左辺を $g(\vec{X})$ と定義すれば、ベイズの識別規則は以下のように書き直せる。

$$\begin{aligned} g(\vec{X}) &\equiv \log \frac{P(\vec{X}|\omega_1)}{P(\vec{X}|\omega_2)} + \log \frac{P(\omega_1)}{P(\omega_2)} \\ \begin{cases} g(\vec{X}) \geq 0 \rightarrow \vec{X} \in \omega_1 \\ g(\vec{X}) < 0 \rightarrow \vec{X} \in \omega_2 \end{cases} \end{aligned}$$

ここで X_1, X_2, \dots, X_r が統計的に独立ならば次式が成り立つ。

$$P(\vec{X}) = P(X_1 \cap X_2 \cap \dots \cap X_r) = P(X_1)P(X_2) \dots P(X_r)$$

$$P(\vec{X} | \omega_i) = P(X_1 | \omega_i)P(X_2 | \omega_i) \dots P(X_r | \omega_i)$$

よって、

$$g(\vec{X}) = \sum_{k=1}^r \log \frac{P(X_k | \omega_1)}{P(X_k | \omega_2)} + \log \frac{P(\omega_1)}{P(\omega_2)}$$

ここで、 $X_k = \{0,1\}$ のとき、

$$P(X_k | \omega_i) = P_k(0 | \omega_i)^{1-X_k} P_k(1 | \omega_i)^{X_k}$$

となる。ただし、 $P_k(0 | \omega_i) \equiv P(X_k = 0 | \omega_i)$ 、 $P_k(1 | \omega_i) \equiv P(X_k = 1 | \omega_i)$ とした。

なぜなら、 $X_k = 0$ のとき、左辺 = $P_k(0 | \omega_i)$ 、右辺 = $P_k(0 | \omega_i)^{1-0} P_k(1 | \omega_i)^0 = P_k(0 | \omega_i)$

$X_k = 1$ のとき、左辺 = $P_k(1 | \omega_i)$ 、右辺 = $P_k(0 | \omega_i)^{1-1} P_k(1 | \omega_i)^1 = P_k(1 | \omega_i)$

よって、

$$\begin{aligned} g(\vec{X}) &= \sum_{k=1}^r \log \frac{P_k(0 | \omega_1)^{1-X_k} P_k(1 | \omega_1)^{X_k}}{P_k(0 | \omega_2)^{1-X_k} P_k(1 | \omega_2)^{X_k}} + \log \frac{P(\omega_1)}{P(\omega_2)} \\ &= \sum_{k=1}^r \left\{ (1-X_k) \log \frac{P_k(0 | \omega_1)}{P_k(0 | \omega_2)} + X_k \log \frac{P_k(1 | \omega_1)}{P_k(1 | \omega_2)} \right\} + \log \frac{P(\omega_1)}{P(\omega_2)} \\ &= \underbrace{\log \frac{P(\omega_1)}{P(\omega_2)} + \sum_{k=1}^r \log \frac{P_k(0 | \omega_1)}{P_k(0 | \omega_2)}}_{w_0} + \sum_{k=1}^r X_k \underbrace{\log \frac{P_k(1 | \omega_1)}{P_k(1 | \omega_2)} \frac{P_k(0 | \omega_1)}{P_k(0 | \omega_2)}}_{w_k} \\ &= \underline{\underline{w_0 + w_1 X_1 + w_2 X_2 + \dots + w_r X_r}} \end{aligned}$$

X の 1 次式 \rightarrow 線形識別関数

2.4 連続分布の識別関数

観測パターン $\vec{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_r \end{bmatrix}$ において、各属性 X_k が任意の実数を取り、観測パターン \vec{X} が

パターンクラス ω_1, ω_2 のいずれかに属する場合も、離散分布の場合と同様にベイズの識別規則は以下ようになる。

$$f(\vec{X}) \equiv P(\omega_1)P(\vec{X} | \omega_1) - P(\omega_2)P(\vec{X} | \omega_2)$$

$$\begin{cases} f(\vec{X}) \geq 0 \rightarrow \vec{X} \in \omega_1 \\ f(\vec{X}) < 0 \rightarrow \vec{X} \in \omega_2 \end{cases}$$

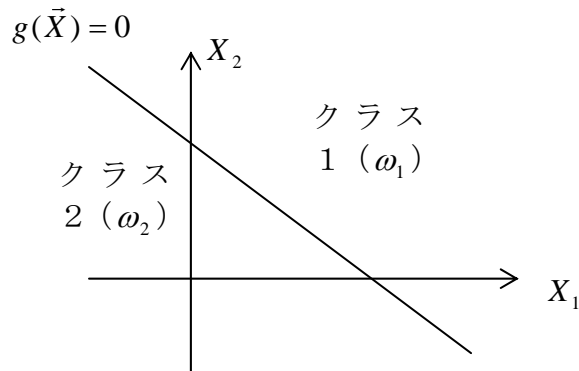
または、

$$g(\vec{X}) \equiv \log \frac{P(\vec{X} | \omega_1)}{P(\vec{X} | \omega_2)} + \log \frac{P(\omega_1)}{P(\omega_2)}$$

$$\begin{cases} g(\vec{X}) \geq 0 \rightarrow \vec{X} \in \omega_1 \\ g(\vec{X}) < 0 \rightarrow \vec{X} \in \omega_2 \end{cases}$$

定理 クラス内のパターン分布が共分散行列の等しい正規分布に従うとき、ベイズの識別規則は線形識別関数になる。(証明は教科書 P.9)

$$g(\vec{X}) = w_0 + w_1 X_1 + w_2 X_2 + \cdots + w_r X_r$$



2.5 正規分布

(1) 1次元正規分布

$$p(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left\{-\frac{1}{2} \frac{(x-\mu)^2}{\sigma^2}\right\}$$

σ^2 : 分散

μ : 平均

(2) 2次元正規分布

$$p(\vec{X}) = \frac{1}{2\pi |\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2} (\vec{X} - \vec{\mu})' \Sigma^{-1} (\vec{X} - \vec{\mu})\right\}$$

(3) n次元正規分布

$$p(\vec{X}) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2} (\vec{X} - \vec{\mu})' \Sigma^{-1} (\vec{X} - \vec{\mu})\right\}$$

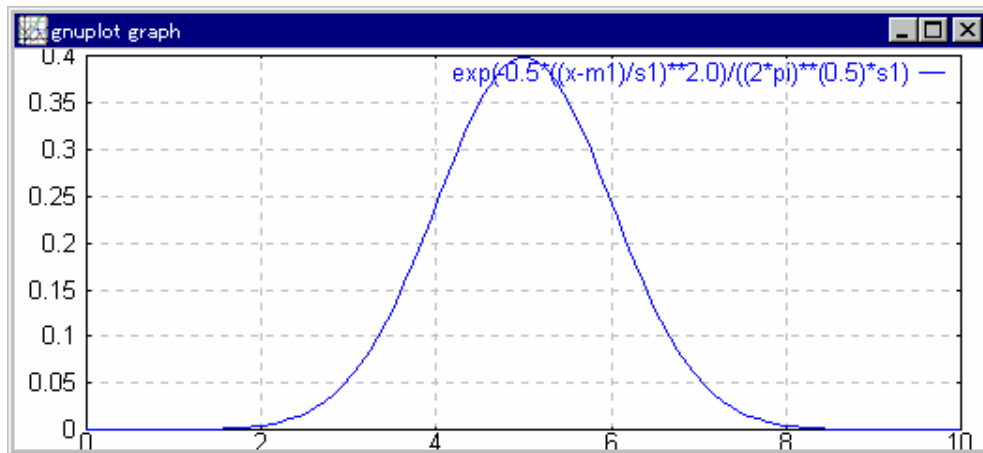
Σ : 共分散行列

(4) 正規分布の例

(a) 平均 5.0, 分散 1.0 の正規分布

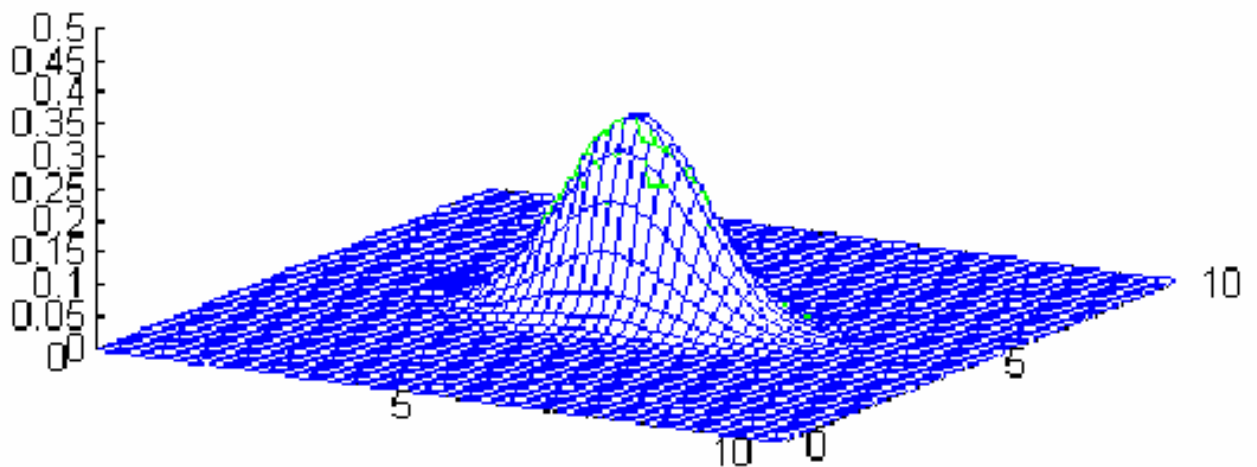
```
[c:¥tmp¥norm. gp]
s1=1.0
m1=5.0
set grid
plot [0:10] exp(-0.5*((x-m1)/s1)**2.0)/((2*pi)**(0.5)*s1)
```

```
gnuplot> cd "c:¥tmp"
gnuplot> load "norm. gp"
```



(b) 平均 $\begin{bmatrix} 5 \\ 5 \end{bmatrix}$, 共分散行列 $\Sigma = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix}$ の正規分布

```
m1=5.0
m2=5.0
s1=1.0
s2=0.5
set grid
set ticslevel 0
set hidden3d
set isosamples 50
splot [0:10][0:10][0:0.5] exp(-0.5*((x-m1)**2.0/s1+(y-m2)**2.0/s2))/(2.0*pi*(s1*s2)**0.5)
```



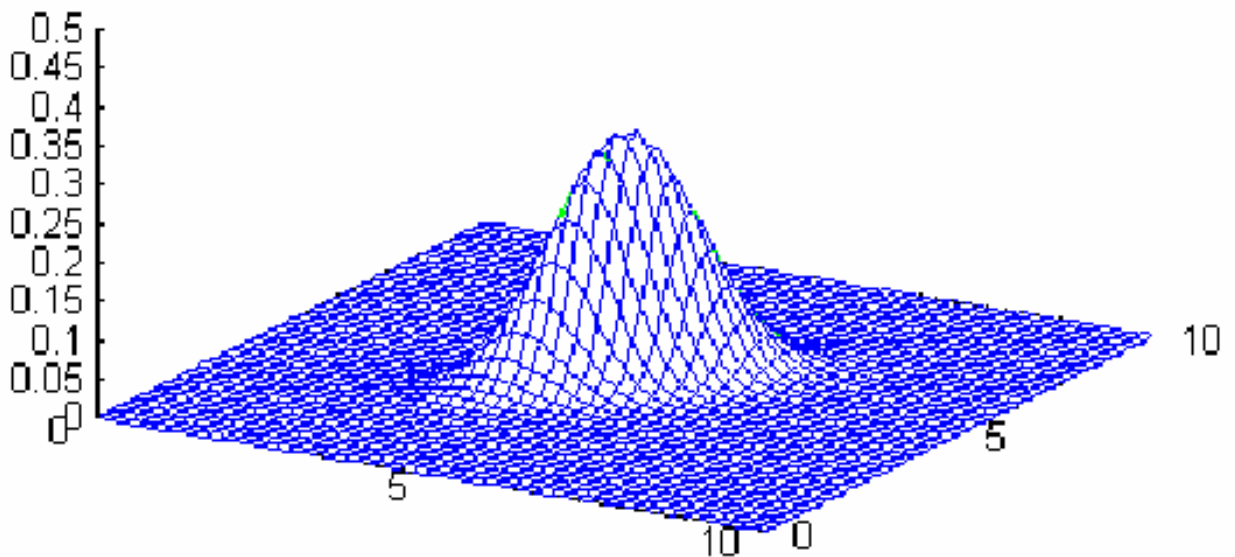
(c) 平均 $\begin{bmatrix} 5 \\ 5 \end{bmatrix}$, 共分散行列 $\Sigma = \begin{bmatrix} 1.0 & 0 \\ 0 & 0.5 \end{bmatrix}$ の正規分布を 45 度回転

平均 $\begin{bmatrix} 5 \\ 5 \end{bmatrix}$, 共分散行列 $\Sigma = \begin{bmatrix} 0.75 & 0.25 \\ 0.25 & 0.75 \end{bmatrix}$, $\Sigma^{-1} = \begin{bmatrix} 1.5 & -0.5 \\ -0.5 & 1.5 \end{bmatrix}$ の正規分布

```
m1=5.0
m2=5.0
s1=1.0
s2=0.5
th=45
```

```
det=(s1*s2)**0.5
s1=1.0/s1
s2=1.0/s2
th=th*pi/180.0
s=sin(th)
c=cos(th)
d1=s1*c*c+s2*s*s
d2=2.0*(s1-s2)*c*s
d3=s1*s*s+s2*c*c
print d1
print d2*0.5
print d3
```

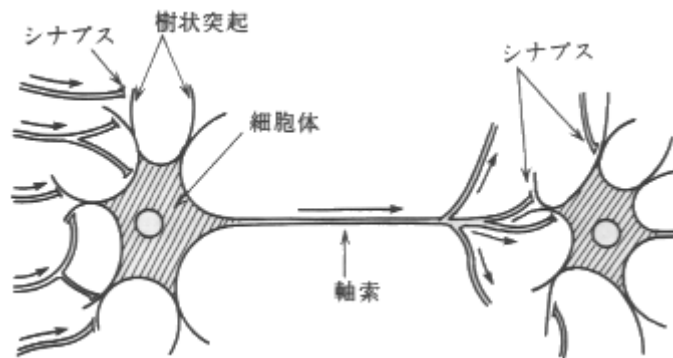
```
set grid
set ticslevel 0
set hidden3d
set isosamples 50
splot [0:10][0:10][0:0.5] exp(-0.5*(d1*(x-m1)**2
+d2*(x-m1)*(y-m2)
+d3*(y-m2)**2))/(2*pi*det)
```



3 ニューラルネットワーク

3.1 人工ニューラルネットワーク

神経細胞（ニューロン）の模式図



人工ニューラル
ネットワーク

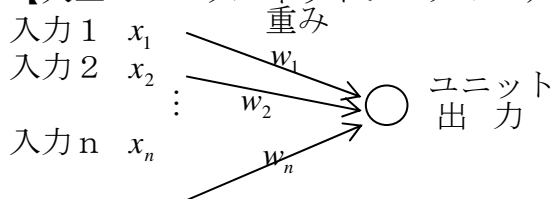
ニューロンが興奮しているとき → 電気パルスは密 ←→
ニューロンが興奮していないとき → 電気パルスは疎 ←→

出力が大きい
出力が小さい

別のニューロンの興奮度をどの程度伝えるか
(シナプスでの Na, K イオン濃度によって調整)

←→ 重み
大きい → 良く伝える
小さい → あまり伝えない

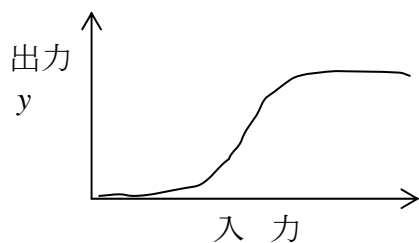
【人工ニューラルネットワークのモデル】



出力 y は、入力 x_1, \dots, x_n 及び 重み w_1, \dots, w_n で決まる。
 $y = f(x_1, \dots, x_n, w_1, \dots, w_n)$

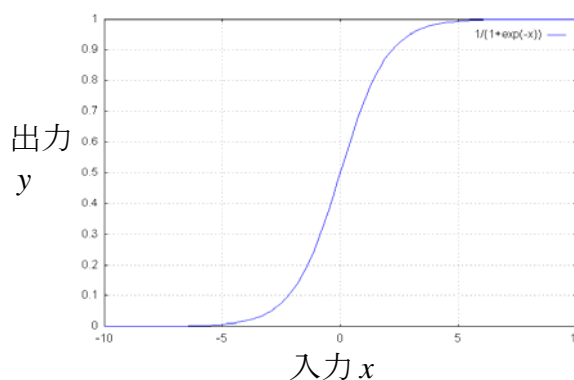
ヤリイカのニューロンによる実験結果

人工ニューラルネットワークでよく使われる関数 (シグモイド関数)

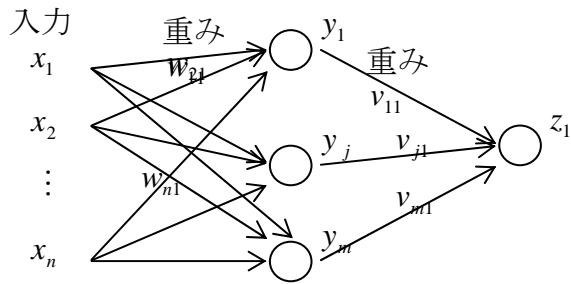


$$f(x) = \frac{1}{1 + \exp(-x)}$$

$$x = w_1 x_1 + w_2 x_2 + \dots + w_n x_n = \sum_{i=1}^n w_i x_i$$



3.2 階層型モデル



1 番目の隠れユニットの出力 $y_1 = f(w_{11}x_1 + w_{21}x_2 + \dots + w_{n1}x_n)$

j 番目の隠れユニットの出力 $y_j = f(w_{1j}x_1 + w_{2j}x_2 + \dots + w_{nj}x_n)$

出力 $z_1 = f(v_{11}y_1 + v_{21}y_2 + \dots + v_{m1}y_m)$

ニューラルネットワークの学習...ある入力に対して望ましい値が出力されるように重み w_{ij}, v_{jl} を決める。

多数の例を用意 → 重みを学習

	入力	出力	望ましい出力
例 1	$x_{11}x_{21} \dots x_{n1} \rightarrow$	$z_{11} \dots$	d_1
例 2	$x_{12}x_{22} \dots x_{n2} \rightarrow$	$z_{12} \dots$	d_2
	\vdots		
例 c	$x_{1c}x_{2c} \dots x_{nc} \rightarrow$	$z_{1c} \dots$	d_c
	\vdots		
例 N	$x_{1N}x_{2N} \dots x_{nN} \rightarrow$	$z_{1N} \dots$	d_N

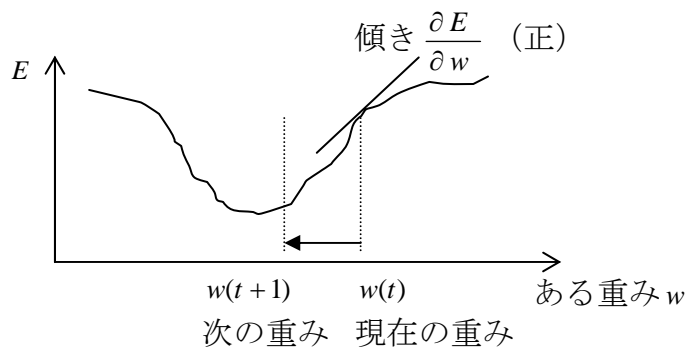
出力と望ましい出力がなるべく近くなるように重みを決めればよい。どの程度出力と望ましい出力が近いかを表すために次式でエラー（誤差）を定義する。

$$\text{エラー (誤差)} E \equiv \frac{1}{2} \sum_{c=1}^N (z_{1c} - d_c)^2$$

出力と望ましい出力の差の 2 乗和をエラーと定義



重み w_{ij}, v_{jl} を変化させてエラー E を最小化すればよい。



傾き $\frac{\partial E}{\partial w}$ が正のとき

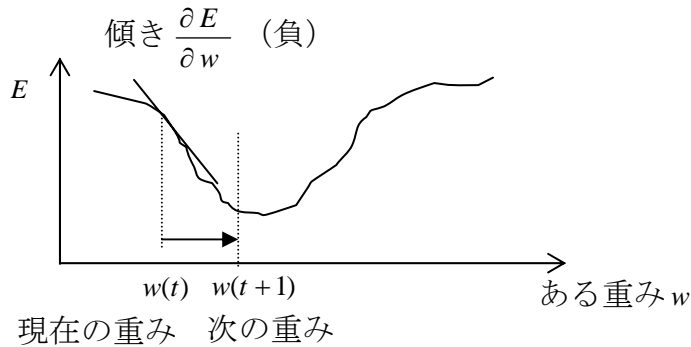
$$w(t+1) = w(t) - \varepsilon \frac{\partial E}{\partial w}$$

新しい重み (負の方向へ移動)

現在の重み

傾き

比例定数



傾き $\frac{\partial E}{\partial w}$ が負のとき

$$w(t+1) = w(t) - \varepsilon \frac{\partial E}{\partial w}$$

新しい重み (正の方向へ移動)

現在の重み

傾き

比例定数

重みの単位時間当たりの変化分 $w(t+1) - w(t) = -\varepsilon \frac{\partial E}{\partial w}$

$$\frac{dw}{dt} = -\varepsilon \frac{\partial E}{\partial w}$$

【定理】 次式を満たせば、エラー E は時間 t に対して非増加である。

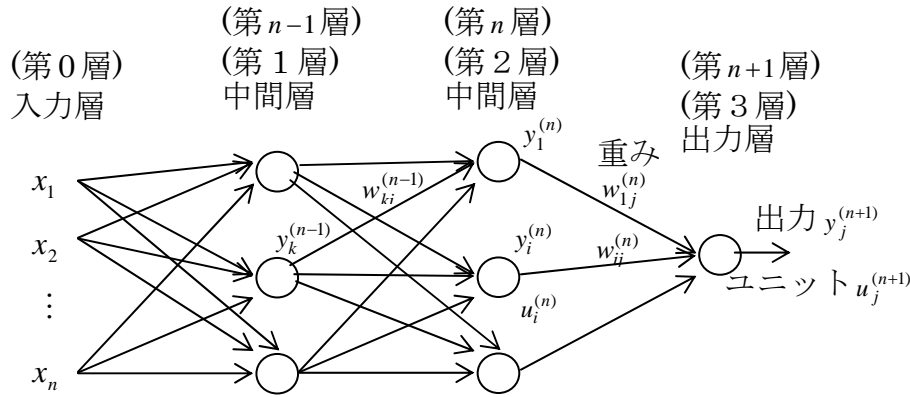
$$\begin{cases} \frac{dv_{j1}}{dt} = -\varepsilon \frac{\partial E}{\partial v_{j1}} & (j = 1, 2, \dots, m) \\ \frac{dw_{ij}}{dt} = -\varepsilon \frac{\partial E}{\partial w_{ij}} & (i = 1, 2, \dots, n; j = 1, 2, \dots, m) \end{cases}$$

(証明) エラー E は重み w_{ij}, v_{j1} の関数なので

$$\begin{aligned} \frac{dE}{dt} &= \sum_{i=1}^n \sum_{j=1}^m \frac{\partial E}{\partial w_{ij}} \frac{dw_{ij}}{dt} + \sum_{j=1}^m \frac{\partial E}{\partial v_{j1}} \frac{dv_{j1}}{dt} \\ &= \sum_{i=1}^n \sum_{j=1}^m \frac{\partial E}{\partial w_{ij}} \left(-\varepsilon \frac{\partial E}{\partial w_{ij}} \right) + \sum_{j=1}^m \frac{\partial E}{\partial v_{j1}} \left(-\varepsilon \frac{\partial E}{\partial v_{j1}} \right) \\ &= -\varepsilon \left\{ \sum_{i=1}^n \sum_{j=1}^m \left(\frac{\partial E}{\partial w_{ij}} \right)^2 + \sum_{j=1}^m \left(\frac{\partial E}{\partial v_{j1}} \right)^2 \right\} \end{aligned}$$

$$\leq 0$$

3.3 バックプロパゲーション



階層型ニューラルネットの学習法として、D.E.Rumelhart らによる **back-propagation** 学習手続きを説明する。第0層を入力層、第1層～第n層を中間層、第n+1層を出力層とし、第n層*i*番目のユニットを $u_i^{(n)}$ と呼ぶことにする。ユニット $u_i^{(n)}$ からユニット $u_j^{(n+1)}$ への接続における重みを $w_{ij}^{(n)}$ 、ユニット $u_i^{(n)}$ の出力を $y_i^{(n)}$ としたとき、ユニット $u_j^{(n+1)}$ への総入力 $x_j^{(n+1)}$ 、ユニット $u_j^{(n+1)}$ の出力 $y_j^{(n+1)}$ を次式のように定義する。

$$x_j^{(n+1)} = \sum_i y_i^{(n)} w_{ij}^{(n)} \tag{1}$$

$$y_j^{(n+1)} = \frac{1}{1 + \exp(-x_j^{(n+1)})} \tag{2}$$

ある望ましい入力・出力の組 c について ユニット $u_j^{(n+1)}$ における実際の出力を $y_{jc}^{(n+1)}$ 、望ましい出力を d_{jc} としたとき、エラー E を次式で定義する。

$$E = \frac{1}{2} \sum_c \sum_j (y_{jc}^{(n+1)} - d_{jc})^2 \tag{3}$$

このエラーが小さいほど有効なネットワークが作成されたことにする。学習は出力層から入力層へ向かって逆方向に進む。各重み w について $\Delta w = -\varepsilon \frac{\partial E}{\partial w}$ を求めれば、重み変化 Δw を次式のように決定できる。

$$\Delta w = -\varepsilon \frac{\partial E}{\partial w} \tag{4}$$

ここで ε は重み変化の度合いを決める係数である。

第n層から第n+1層への重みについて、(4)式の $\frac{\partial E}{\partial w}$ は、以下のように計算される。まず(3)式より、エラー E は $y_{jc}^{(n+1)}$ の関数である。また(2)式より $y_{jc}^{(n+1)}$ は $x_{jc}^{(n+1)}$ の関数なので、エラー E は $x_{jc}^{(n+1)}$ の関数である。よって、

$$\frac{\partial E}{\partial w_{ij}^{(n)}} = \sum_c \sum_m \frac{\partial E}{\partial x_{mc}^{(n+1)}} \frac{\partial x_{mc}^{(n+1)}}{\partial w_{ij}^{(n)}} \tag{5}$$

となる。(1)式より $x_{mc}^{(n+1)}$ の式には $w_{mi}^{(n)}$ しか含まれていない。よって、 $\frac{\partial x_{mc}^{(n+1)}}{\partial w_{ij}^{(n)}}$ は $m=j$ のとき

$y_{ic}^{(n)}$ 、それ以外の場合は0なので、

$$\frac{\partial E}{\partial w_{ji}^{(n)}} = \sum_c \frac{\partial E}{\partial x_{jc}^{(n+1)}} \frac{\partial x_{jc}^{(n+1)}}{\partial w_{ji}^{(n)}} = \sum_c \frac{\partial E}{\partial x_{jc}^{(n+1)}} y_{ic}^{(n)} \tag{6}$$

となる。(2)式より $x_{jc}^{(n+1)}$ は $y_{jc}^{(n+1)}$ と 1 対 1 に対応するので、 $\frac{\partial E}{\partial x_{jc}^{(n+1)}} = \frac{\partial E}{\partial y_{jc}^{(n+1)}} \frac{dy_{jc}^{(n+1)}}{dx_{jc}^{(n+1)}}$ となる。

さらに(2),(3)式より

$$\frac{\partial E}{\partial x_{jc}^{(n+1)}} = (y_{jc}^{(n+1)} - d_{jc})y_{jc}^{(n+1)}(1 - y_{jc}^{(n+1)}) \quad (7)$$

なので、 $\frac{\partial E}{\partial w_{ji}^{(n)}}$ が次式のように求まる。

$$\frac{\partial E}{\partial w_{ji}^{(n)}} = \sum_c (y_{jc}^{(n+1)} - d_{jc})y_{jc}^{(n+1)}(1 - y_{jc}^{(n+1)})y_{ic}^{(n)} \quad (8)$$

第 $n-1$ 層から第 n 層への重みについて(4)式の $\frac{\partial E}{\partial w}$ は以下のように計算される。(3)式より、エラー E は $y_{jc}^{(n+1)}$ の関数、(2)式より $y_{jc}^{(n+1)}$ は $x_{jc}^{(n+1)}$ の関数、(1)式より $x_{jc}^{(n+1)}$ は $y_{ic}^{(n)}$ の関数、(2)式より $y_{ic}^{(n)}$ は $x_{ic}^{(n)}$ の関数である。よって、エラー E は $x_{ic}^{(n)}$ の関数であるので、

$$\frac{\partial E}{\partial w_{ki}^{(n-1)}} = \sum_c \sum_m \frac{\partial E}{\partial x_{mc}^{(n)}} \frac{\partial x_{mc}^{(n)}}{\partial w_{ki}^{(n-1)}} \quad (9)$$

となる。(1)式より $x_{mc}^{(n)}$ の式には $w_{km}^{(n-1)}$ しか含まれていない。よって、 $\frac{\partial x_{mc}^{(n)}}{\partial w_{ki}^{(n-1)}}$ は $m=i$ のとき $y_{kc}^{(n-1)}$ 、それ以外の場合は 0 なので、

$$\frac{\partial E}{\partial w_{ki}^{(n-1)}} = \sum_c \frac{\partial E}{\partial x_{ic}^{(n)}} \frac{\partial x_{ic}^{(n)}}{\partial w_{ki}^{(n-1)}} = \sum_c \frac{\partial E}{\partial x_{ic}^{(n)}} y_{kc}^{(n-1)} \quad (10)$$

となる。(2)式より $x_{ic}^{(n)}$ は $y_{ic}^{(n)}$ と 1 対 1 に対応するので次式のように変形できる。

$$\frac{\partial E}{\partial w_{ki}^{(n-1)}} = \sum_c \frac{\partial E}{\partial y_{ic}^{(n)}} \frac{dy_{ic}^{(n)}}{dx_{ic}^{(n)}} y_{kc}^{(n-1)} \quad (11)$$

(2),(3)式より

$$\begin{aligned} \frac{\partial E}{\partial w_{ki}^{(n-1)}} &= \sum_c \frac{\partial E}{\partial y_{ic}^{(n)}} y_{ic}^{(n)} (1 - y_{ic}^{(n)}) y_{kc}^{(n-1)} \\ &= \sum_c \left(\sum_j \frac{\partial E}{\partial x_{jc}^{(n+1)}} \frac{\partial x_{jc}^{(n+1)}}{\partial y_{ic}^{(n)}} \right) y_{ic}^{(n)} (1 - y_{ic}^{(n)}) y_{kc}^{(n-1)} \\ &= \sum_c \left(\sum_j \frac{\partial E}{\partial x_{jc}^{(n+1)}} w_{ij}^{(n)} \right) y_{ic}^{(n)} (1 - y_{ic}^{(n)}) y_{kc}^{(n-1)} \end{aligned} \quad (12)$$

となる。ここで、 $\frac{\partial E}{\partial x_{jc}^{(n+1)}}$ は、前段階の(7)式で求まっている。このように 1 つ後段(出力側)の結果を利用しながら、入力側に向けて学習が進むため、逆伝搬(back-propagation)学習と呼ばれている。

以上のように $\frac{\partial E}{\partial w}$ を求めることができ、(4)式より重み変化 Δw が計算される。この方法は簡単であり、並列計算が可能である。この簡単さと並列計算の可能性をいかしつつ学習時間を短縮するため、指数的減衰係数 α を導入して(4)式を次式のように変更する。これによってネットワークは継続して入力するベクトルに対して似た表現を使用するようになり、学習時間が短縮される。

$$\Delta w(t) = -\varepsilon \frac{\partial E}{\partial w(t)} + \alpha \Delta w(t-1) \quad (13)$$

4 DP マッチング (動的計画法)

4.1 パターンの長さが同じもの同士の比較

例えば、

パターン 1	x_1	x_2	x_3
	\downarrow	\downarrow	\downarrow
パターン 2	y_1	y_2	y_3

とすれば、パターン 1 とパターン 2 との間の距離を

$$\text{パターン 1 とパターン 2 との間の距離} = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2 + (x_3 - y_3)^2}$$

と定義できる。

4.2 パターンの長さが異なるもの同士の比較

例えば、

パターン A	a_1	a_2	a_3	a_4	a_5	
パターン B	b_1	b_2	b_3	b_4	b_5	b_6

とすれば、パターン A とパターン B との間の距離を $D(A, B)$ で表したとき、以下の関係を満たす必要がある。

- $D(A, B) \geq 0$
- パターン A とパターン B が類似している $\leftrightarrow D(A, B)$ の値が小さい

以上の関係を満たす一例としてパターン B の長さをパターン A の長さに合わせて、

4.1 節と同様にパターン A とパターン B との間の距離を $D(A, B)$ を求めてみる。

パターン B	b_1	b_2	b_3	b_4	b_5	b_6	長さ 6
対応 $w(i)$	1	1	2	3	6		$\downarrow b'_i = b_{w(i)}$
$b_{w(i)}$	b_1	b_1	b_2	b_3	b_6		長さ 5
パターン B'	b'_1	b'_2	b'_3	b'_4	b'_5		長さ 5
	\downarrow	\downarrow	\downarrow	\downarrow	\downarrow		
パターン A	a_1	a_2	a_3	a_4	a_5		長さ 5 (I)

パターン A とパターン B との対応 $w(i)$ が与えられているときのパターン間距離 $D(A, B; w)$ を、

$$D(A, B; w) \equiv \sum_{i=1}^I d(a_i, b'_i) = \sum_{i=1}^I d(a_i, b_{w(i)})$$

と定義できる。ここで、 $d(a_i, b'_i)$ は、 a_i と b'_i の距離、 I はパターン A の長さを表す。

パターン A とパターン B との対応 $w(i)$ が不明のとき、パターン A とパターン B とのパターン間距離 $D(A, B)$ を

$$D(A, B) \equiv \min_w D(A, B; w)$$

と定義できる。すなわち、全ての対応 w の中で最も距離が小さくなるときの距離をパターン A とパターン B とのパターン間距離 $D(A, B)$ と定義する。

しかし、計算量が非常に大きい。パターンAの長さをI、パターンBの長さをJとしたとき、加算及び比較回数は以下ようになる。

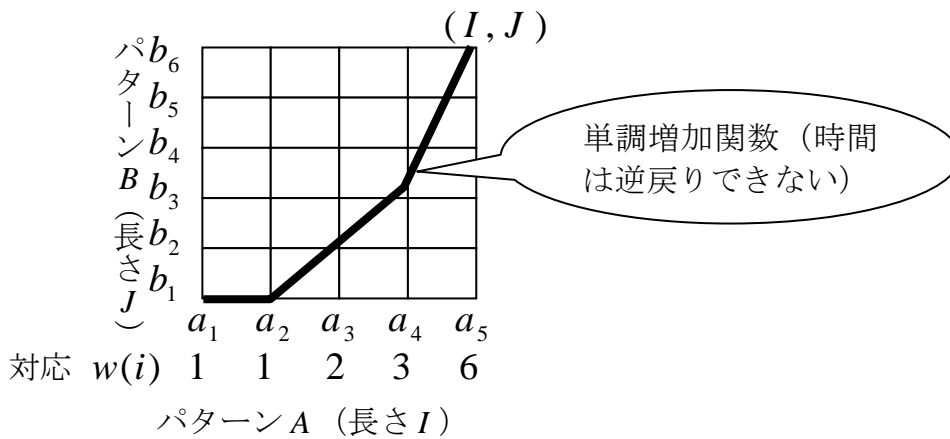
$$\begin{cases} \text{加算回数} = (I-1) \frac{(I+J-3)!}{(I-2)!(J-1)!} \\ \text{比較回数} = \frac{(I+J-3)!}{(I-2)!(J-1)!} - 1 \end{cases}$$

例えば、 $I=25, J=35$ のとき、加算 1.2×10^{17} 回、比較 5.3×10^{15} 回となる。1秒間に 10^{10} 回の加算が実行できる超高速計算機をもってしても、加算だけで 1.2×10^7 秒 (約 139 日) を要する。

4.3 DP マッチング

Dynamic Programming … 既に計算した途中結果をうまく再利用して効率的に計算する方法 (漸化式、再帰を利用)

4.2 節のパターンAとパターンBとの対応 $w(i)$ を図示すると以下ようになる。



パターンAとパターンBとの対応 w が与えられたとき、(1,1) から (i, j) まで累積距離を

$$g(i, j; w) \equiv \sum_{k=1}^i d(a_k, b_{w(k)})$$

と定義する。また、(1,1) から (i, j) まで最小の累積距離を

$$g(i, j) \equiv \min_w g(i, j; w)$$

と定義できる。ここで、 $g(I, J)$ は、(1,1) から (I, J) までの最小の累積距離であるから、パターンAとパターンBとのパターン間距離 $D(A, B)$ と等しい。

$$D(A, B) = g(I, J)$$

よって、 $g(I, J)$ を漸化式を用いて効率的に求めることにする。

- $i=1$ のとき

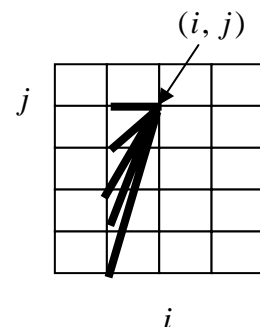
$$g(1, 1) = d(a_1, b_1)$$

- $i=2$ のとき、 $j=1, 2, \dots, J$ について

$$g(2, j) = d(a_1, b_1) + d(a_2, b_j) = g(1, 1) + d(a_2, b_j)$$

- $i=3, 4, \dots, I$ のとき、 $j=1, 2, \dots, J$ について

$$g(i, j) = \min\{g(i-1, 1), g(i-1, 2), \dots, g(i-1, j)\} + d(a_i, b_j)$$



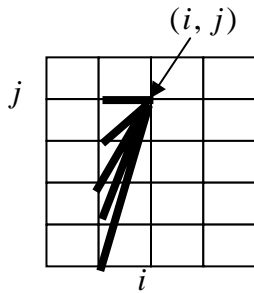
これは、 $g(i-1, j)$ から $g(i, j)$ を求める漸化式になっている。
 以上の方法の計算量は以下ようになる。

$$\begin{cases} \text{加算回数} = (I-1)J \\ \text{比較回数} = (I-1)(J-1)J/2 \end{cases}$$

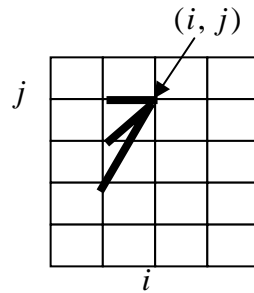
例えば、 $I=25, J=35$ のとき、加算840回、比較14,280回となる。1秒間に 10^{10} 回の加算が実行できる計算機なら、 8.4×10^{-8} 秒（84ns）で計算が終了する。

4.4 傾斜制限

場合によっては、あまり極端な対応（伸縮）をしない方がよい。

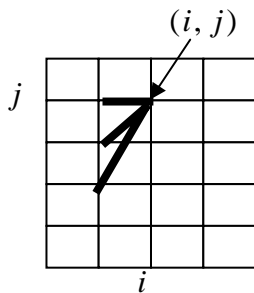


傾斜制限しない場合

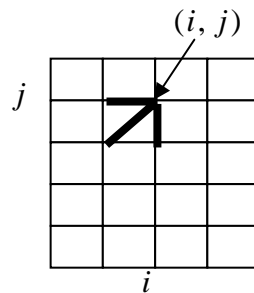


傾斜制限した場合

4.5 対称な対応



非対称な対応
 $D(A, B) \neq D(B, A)$



対称な対応
 $D(A, B) = D(B, A)$

4.6 脱落と挿入 (教科書 P.100 参照)

パターン A = $a_0 \ a_1 \ a_2 \ \dots \ a_I$ (長さ I+1)
 $\alpha_0 \ \alpha_1 \ \alpha_2 \ \dots \ \alpha_K$ (長さ K+1)
 (K+1)個に長さをあわせる
 $\beta_0 \ \beta_1 \ \beta_2 \ \dots \ \beta_K$ (長さ K+1)
 パターン B = $b_0 \ b_1 \ b_2 \ \dots \ b_J$ (長さ J+1)

P.99 の例の場合

A = $a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5$
 $\alpha(P_k) = a_0 \ a_0^* \ a_1 \ a_2 \ a_3 \ a_3^* \ a_3^* \ a_4 \ a_5$
 $P_0 \ P_1 \ P_2 \ P_3 \ P_4 \ P_5 \ P_6 \ P_7 \ P_8$
 $\beta(P_k) = b_0 \ b_1 \ b_2 \ b_2^* \ b_3 \ b_4 \ b_5 \ b_6 \ b_6^*$
 B = $b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5 \ b_6$

上の例ではパターン A が長さ 6、パターン B が長さ 7 である。パターン A を $\alpha(P_k)$ 、パターン B を $\beta(P_k)$ のように変更すると、両パターンとも長さ 9 となり、パターン間距離を求めることができる。ここで $P_k = (i(k), j(k))$ は、両パターンの k 番目の対応点で、 $a_{i(k)}$ と $b_{j(k)}$ が対応することを示している。以上の対応は、右図のようにも表せる。

上の例では一つの a_0 に 2 つの $b_0 \ b_1$ を対応させている。しかし、パターンによっては、 a_0 と b_0 が対応し、 b_1 に対応するものが脱落していると考え方が自然な場合がある。脱落していると考えられる所を * で表してみると、

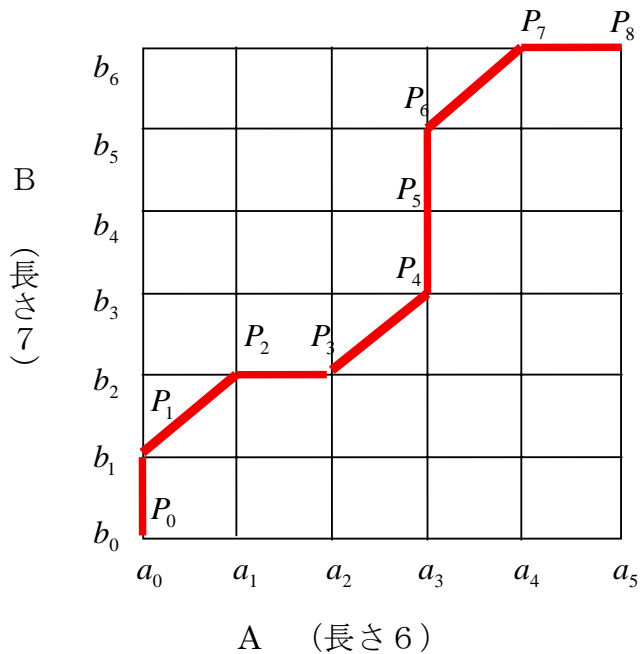
$\alpha(P_k) = a_0 \ * \ a_1 \ a_2 \ a_3 \ * \ * \ a_4 \ a_5$
 $\beta(P_k) = b_0 \ b_1 \ b_2 \ * \ b_3 \ b_4 \ b_5 \ b_6 \ *$

となる。ここで、

$\alpha(P_k) \equiv \begin{cases} a_{i(k)} & i(k) = i(k-1) + 1 \text{ のとき} \\ * & i(k) = i(k-1) \text{ のとき} \end{cases}$

$\beta(P_k) \equiv \begin{cases} b_{j(k)} & j(k) = j(k-1) + 1 \text{ のとき} \\ * & j(k) = j(k-1) \text{ のとき} \end{cases}$

である。よって、



→ ↗ 方向に移動している時

↑ 方向に移動している時

↗ ↑ 方向に移動している時

→ 方向に移動している時

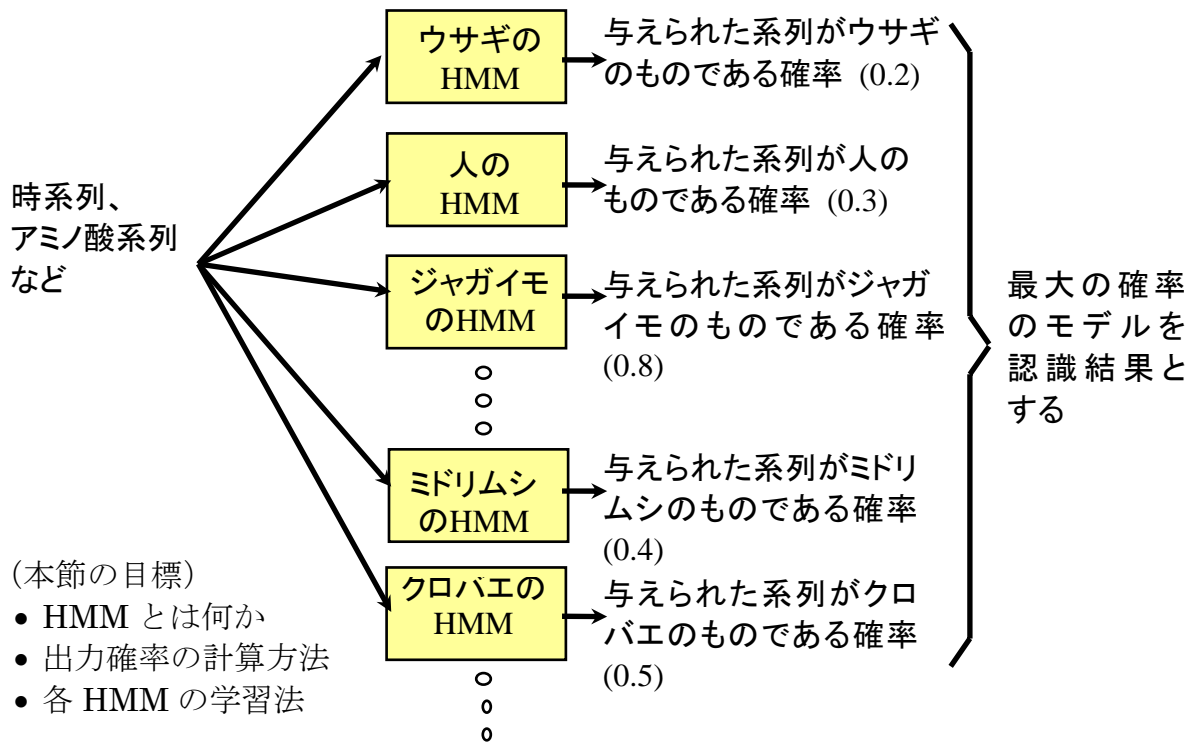
$$D(A, B) \equiv \min_{P_k} \sum_{k=0}^K d(\alpha(p_k), \beta(P_k))$$

(計算方法)

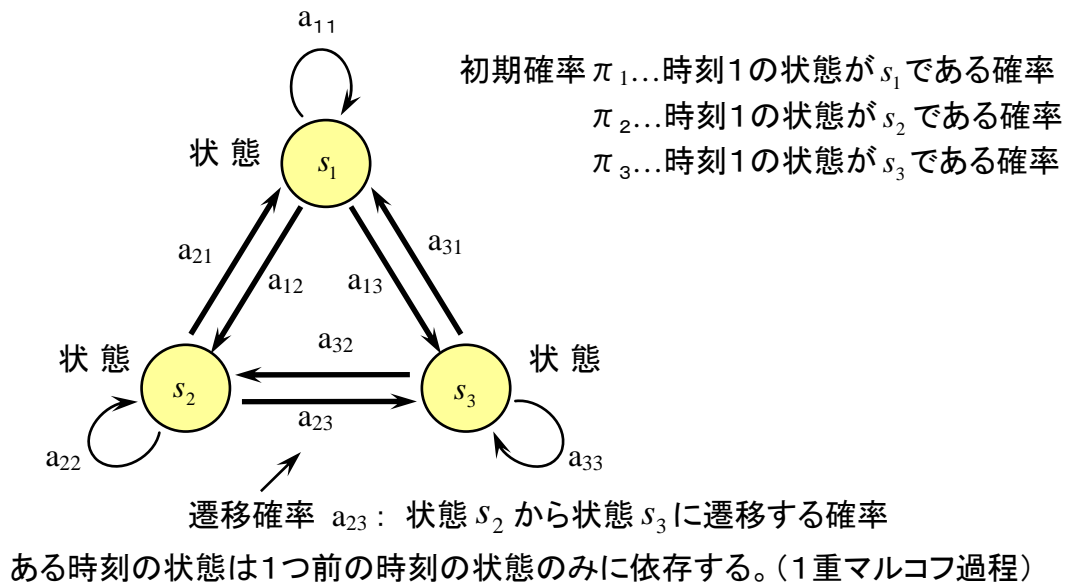
$$\begin{aligned} \textcircled{1} \quad & g(0,0) = d(a_0, b_0) \\ & g(i,0) = g(i-1,0) + d(a_i, *) \quad \rightarrow \text{方向 } (i=1,2,\dots,I) \\ \textcircled{2} \quad & g(0,j) = g(0,j-1) + d(*, b_j) \quad \uparrow \text{方向 } (j=1,2,\dots,J) \\ \textcircled{3} \quad & g(i,j) = \min \left\{ \begin{array}{l} g(i-1, j) + d(a_i, *) \\ g(i-1, j-1) + d(a_i, b_j) \\ g(i, j-1) + d(*, b_j) \end{array} \right\} \end{aligned}$$

5 HMM (Hidden Markov Model)

5.1 HMM を用いたパターン認識例



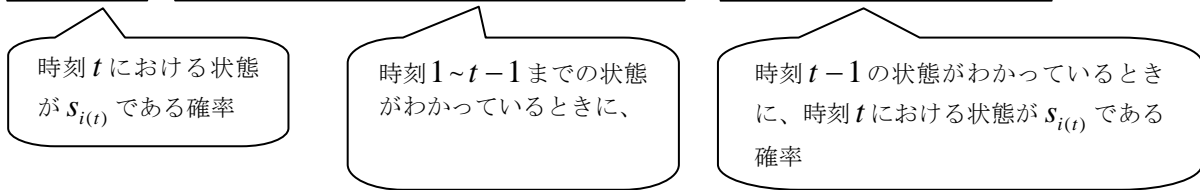
5.2 マルコフモデル



(5) マルコフモデルとは

今、有限個 (n 個) の状態の集合 $\{s_1, s_2, \dots, s_n\}$ があり、時刻 t における状態を S_t で表わされるとする。このとき、次式が成立するモデルを (1 重) マルコフモデルと呼ぶ。

$$P(S_t = s_{i(t)} | S_1 = s_{i(1)} \cap S_2 = s_{i(2)} \cap \dots \cap S_{t-1} = s_{i(t-1)}) = P(S_t = s_{i(t)} | S_{t-1} = s_{i(t-1)})$$



すなわち、

- 1つ前の時刻 ($t-1$) の状態 S_{t-1} のみによって、次の時刻 (t) の状態 S_t がいずれの状態になるかという確率が決まるもの → (1 重) マルコフモデル
- 1つ前の時刻 ($t-1$) の状態 S_{t-1} と 2つ前の時刻 ($t-2$) の状態 S_{t-2} によって、次の時刻 (t) の状態 S_t がいずれの状態になるかという確率が決まるもの → 2 重マルコフモデル

(6) マルコフモデルを決めるパラメータ

初期分布 $\Pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{bmatrix} = \begin{bmatrix} P(S_1 = s_1) \\ P(S_1 = s_2) \\ \vdots \\ P(S_1 = s_N) \end{bmatrix}$ 時刻 1 における状態確率 (状態数を N とする)

遷移確率 $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}$

時刻 t において状態 s_i から s_j に遷移する確率 $a_{ij}(t) \equiv P(S_{t+1} = s_j | S_t = s_i)$

時刻に無関係 (一様なマルコフ連鎖)

$a_{ij}(t) = a_{ij}$

時刻 1 ~ t までの状態が $s_{i(1)}, s_{i(2)}, \dots, s_{i(t)}$ である確率

$$P(S_t = s_{i(t)} \cap S_{t-1} = s_{i(t-1)} \cap \dots \cap S_1 = s_{i(1)})$$

A

B

$$P(A \cap B) = P(A | B)P(B)$$

$$= P(S_t = s_{i(t)} | S_{t-1} = s_{i(t-1)} \cap \dots \cap S_1 = s_{i(1)}) P(S_{t-1} = s_{i(t-1)} \cap \dots \cap S_1 = s_{i(1)})$$

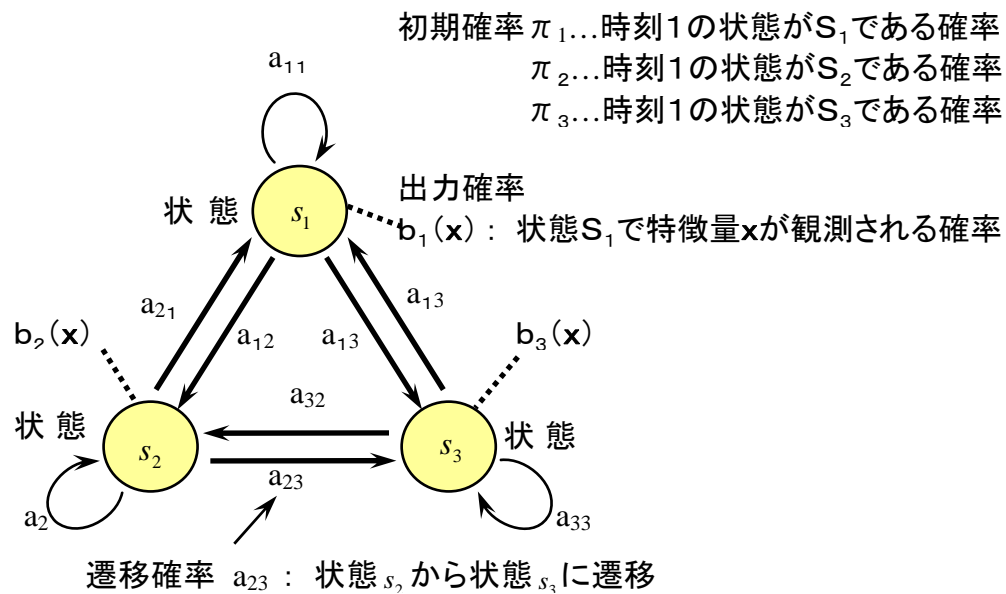
(1 重マルコフモデルの定義より 1つ前の時刻の状態のみによって、次の時刻の状態確率が決まる)

$$\begin{aligned}
 &= P(S_t = s_{i(t)} | S_{t-1} = s_{i(t-1)})P(S_{t-1} = s_{i(t-1)} \cap \dots \cap S_1 = s_{i(1)}) \\
 &= a_{i(t-1)i(t)} P(S_{t-1} = s_{i(t-1)} \cap \dots \cap S_1 = s_{i(1)}) \\
 &= a_{i(t-1)i(t)} P(S_{t-1} = s_{i(t-1)} | S_{t-2} = s_{i(t-2)} \cap \dots \cap S_1 = s_{i(1)})P(S_{t-2} = s_{i(t-2)} \cap \dots \cap S_1 = s_{i(1)}) \\
 &= a_{i(t-1)i(t)} a_{i(t-2)i(t-1)} P(S_{t-2} = s_{i(t-2)} \cap \dots \cap S_1 = s_{i(1)}) \\
 &= a_{i(t-1)i(t)} a_{i(t-2)i(t-1)} \dots a_{i(1)i(2)} P(S_1 = s_{i(1)}) \\
 &= a_{i(t-1)i(t)} a_{i(t-2)i(t-1)} \dots a_{i(1)i(2)} \pi_{i(1)}
 \end{aligned}$$

5.3 Hidden Markov Model

(7) Hidden Markov Model とは

マルコフモデル+各状態で各記号を出力する確率を規定



ある時刻の状態は1つ前の時刻の状態のみに依存する。(1重マルコフ過程)
 ある時刻の出力確率は、その時刻の状態のみに依存する。

【例 1】 ある状態 s_i で

$$\begin{bmatrix} \text{記号 } o_1 \text{ を出力する確率 } 0.2 \\ \text{記号 } o_2 \text{ を出力する確率 } 0.3 \\ \vdots \\ \text{記号 } o_M \text{ を出力する確率 } 0.1 \end{bmatrix}$$

【例 2】

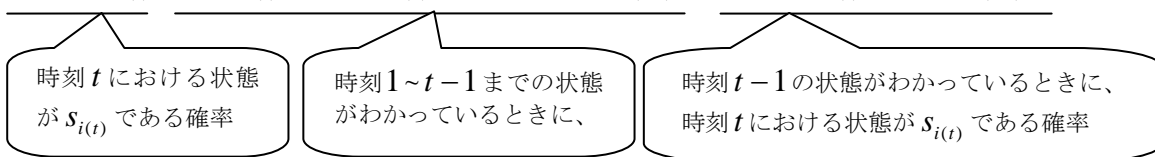
時刻	1	2	3	4	...
状態	s_2	s_3	s_1	s_3	...
出力確率	$\begin{bmatrix} o_1 : 0.2 \\ o_2 : 0.3 \\ \vdots \\ o_M : 0.1 \end{bmatrix}$	$\begin{bmatrix} o_1 : 0.1 \\ o_2 : 0.2 \\ \vdots \\ o_M : 0.5 \end{bmatrix}$	$\begin{bmatrix} o_1 : 0.3 \\ o_2 : 0.1 \\ \vdots \\ o_M : 0.2 \end{bmatrix}$	$\begin{bmatrix} o_1 : 0.1 \\ o_2 : 0.2 \\ \vdots \\ o_M : 0.5 \end{bmatrix}$	

出力が確率でしか規定されない→Hidden

(8) Hidden Markov Model の定義

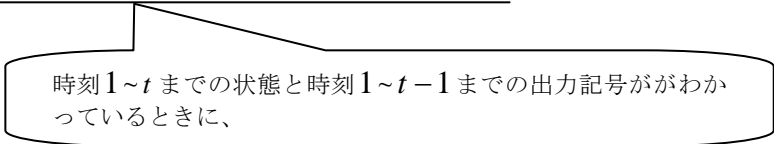
【定義 1】 1つ前の時刻 ($t-1$) の状態 S_{t-1} のみによって、次の時刻 (t) の状態 S_t がいずれの状態になるかという確率が決まる。(マルコフモデルと同じ)

$$P(S_t = s_{i(t)} | S_1 = s_{i(1)} \cap S_2 = s_{i(2)} \cap \dots \cap S_{t-1} = s_{i(t-1)}) = P(S_t = s_{i(t)} | S_{t-1} = s_{i(t-1)})$$



【定義 2】 出力確率はその時刻の状態のみによって決まる。

$$P(O_t = o_{k(t)} | S_1 = s_{i(1)} \cap O_1 = o_{k(1)} \cap S_2 = s_{i(2)} \cap O_2 = o_{k(2)} \cap \dots \cap S_{t-1} = s_{i(t-1)} \cap O_{t-1} = o_{k(t-1)} \cap S_t = s_{i(t)}) = P(O_t = o_{k(t)} | S_t = s_{i(t)})$$



(9) Hidden Markov Model のパラメータ

初期分布 $\Pi = \begin{bmatrix} \pi_1 \\ \pi_2 \\ \vdots \\ \pi_N \end{bmatrix} = \begin{bmatrix} P(S_1 = s_1) \\ P(S_1 = s_2) \\ \vdots \\ P(S_1 = s_N) \end{bmatrix}$ 時刻 1 における状態確率 (状態数を N とする)

遷移確率 $A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N} \\ a_{21} & a_{22} & \dots & a_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{N1} & a_{N2} & \dots & a_{NN} \end{bmatrix}$

時刻 t において状態 s_i から s_j に遷移する確率

$$a_{ij}(t) \equiv P(S_{t+1} = s_j | S_t = s_i)$$

↓ 時刻に無関係 (一様なマルコフ連鎖)

$$a_{ij}(t) = a_{ij}$$

シンボル (記号)

$$\text{出力確率 } B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1M} \\ b_{21} & b_{22} & \cdots & b_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ b_{N1} & b_{N2} & \cdots & b_{NM} \end{bmatrix}$$

状態

時刻 $1 \sim t$ までの状態が $s_{i(1)}, s_{i(2)}, \dots, s_{i(t)}$ で、時刻 $1 \sim t$ までの出力記号が $o_{k(1)}, o_{k(2)}, \dots, o_{k(t)}$ ある確率

$$P(S_1 = s_{i(1)} \cap O_1 = o_{k(1)} \cap S_2 = s_{i(2)} \cap O_2 = o_{k(2)} \cap \cdots \cap S_{t-1} = s_{i(t-1)} \cap O_{t-1} = o_{k(t-1)} \cap S_t = s_{i(t)} \cap O_t = o_{k(t)})$$

$$= P(S_1 = s_{i(1)})P(O_1 = o_{k(1)} \cap S_2 = s_{i(2)} \cap O_2 = o_{k(2)} \cap \cdots \cap S_t = s_{i(t)} \cap O_t = o_{k(t)} | S_1 = s_{i(1)})$$

$$= \pi_{i(1)}P(O_1 = o_{k(1)} \cap S_2 = s_{i(2)} \cap O_2 = o_{k(2)} \cap \cdots \cap S_t = s_{i(t)} \cap O_t = o_{k(t)} | S_1 = s_{i(1)})$$

$$= \pi_{i(1)}P(O_1 = o_{k(1)} | S_1 = s_{i(1)})P(S_2 = s_{i(2)} \cap O_2 = o_{k(2)} \cap \cdots \cap S_t = s_{i(t)} \cap O_t = o_{k(t)} | S_1 = s_{i(1)} \cap O_1 = o_{k(1)})$$

$$= \pi_{i(1)}b_{i(1)k(1)}P(S_2 = s_{i(2)} | S_1 = s_{i(1)} \cap O_1 = o_{k(1)})$$

$$P(O_2 = o_{k(2)} \cap \cdots \cap S_t = s_{i(t)} \cap O_t = o_{k(t)} | S_2 = s_{i(2)} \cap S_1 = s_{i(1)} \cap O_1 = o_{k(1)})$$

(1重マルコフモデルの定義より 1つ前の時刻の状態のみによって、次の時刻の状態確率が決まる)

(HMMの定義より出力確率はその時刻の状態のみによって決まる。)

$$= \pi_{i(1)}b_{i(1)k(1)}P(S_2 = s_{i(2)} | S_1 = s_{i(1)})P(O_2 = o_{k(2)} \cap \cdots \cap S_t = s_{i(t)} \cap O_t = o_{k(t)} | S_2 = s_{i(2)})$$

$$= \pi_{i(1)}b_{i(1)k(1)}a_{i(1)i(2)}P(O_2 = o_{k(2)} \cap \cdots \cap S_t = s_{i(t)} \cap O_t = o_{k(t)} | S_2 = s_{i(2)})$$

$$\vdots$$

$$= \pi_{i(1)}b_{i(1)k(1)}a_{i(1)i(2)}b_{i(2)k(2)} \cdots a_{i(t-1)i(t)}b_{i(t)k(t)}$$

【例】HMMの状態集合 Φ 、出力集合 O 、初期分布 Π 、状態遷移確率 A 、出力確率 B が以下のように表されるとする。

$$\Phi = \{s_1, s_2\}, \quad O = \{o_1, o_2, o_3\},$$

$$\Pi = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix}, \quad A = \begin{bmatrix} 0.7 & 0.3 \\ 0.2 & 0.8 \end{bmatrix}, \quad B = \begin{bmatrix} 0.7 & 0.2 & 0.1 \\ 0.2 & 0.3 & 0.5 \end{bmatrix}$$

(1) 時刻 1 で o_2 、時刻 2 で o_1 を出力し、時刻 1 における状態 S_1 が s_2 、時刻 2 における状態 S_2 が s_1 である確率 $P(O_1 = o_2 \cap O_2 = o_1 \cap S_1 = s_2 \cap S_2 = s_1)$ を求めよ。

(2) $P(O_1 = o_2 \cap O_2 = o_1 \cap S_2 = s_1)$ を求めよ。

(3) $P(O_1 = o_2 \cap O_2 = o_1)$ を求めよ。

(10) 前向きアルゴリズム

HMM の状態集合 $\Phi = \{s_1, s_2, \dots, s_N\}$ 、出力集合 $O = \{o_1, o_2, \dots, o_M\}$ 、初期分布 Π 、状態遷移確率 A 、出力確率 B 、パラメータ $\lambda = (\Pi, A, B)$ とする。このモデルより時刻 $1, 2, \dots, T$ に $o_{k(1)}, o_{k(2)}, \dots, o_{k(T)}$ が出力される確率は次式のようなになる。

$$\begin{aligned} & P_\lambda(O_1 = o_{k(1)} \cap O_2 = o_{k(2)} \cap \dots \cap O_T = o_{k(T)}) \\ &= \sum_{i(1)=1}^N \sum_{i(2)=1}^N \dots \sum_{i(T)=1}^N P(S_1 = s_{i(1)} \cap O_1 = o_{k(1)} \cap S_2 = s_{i(2)} \cap O_2 = o_{k(2)} \cap \dots \cap S_T = s_{i(T)} \cap O_T = o_{k(T)}) \\ &= \sum_{i(1)=1}^N \sum_{i(2)=1}^N \dots \sum_{i(T)=1}^N \pi_{i(1)} b_{i(1)k(1)} a_{i(1)i(2)} b_{i(2)k(2)} \dots a_{i(t-1)i(t)} b_{i(t)k(t)} \end{aligned}$$

この式は N^T 回の積和演算が必要であるため、計算量が膨大である。そこで、以下のように漸化式を用いて効率的に計算を行う（動的計画法）。

まず、途中結果を以下のように定義する。

時刻 $1, 2, \dots, T$ に $o_{k(1)}, o_{k(2)}, \dots, o_{k(t)}$ を出力し、時刻 t における状態が s_j である確率

$$\alpha_t(j) \equiv P_\lambda(O_1 = o_{k(1)} \cap O_2 = o_{k(2)} \cap \dots \cap O_t = o_{k(t)} \cap S_t = s_j)$$

【漸化式】

・ 初期条件 $t=1$ のとき $\alpha_1(j) = P_\lambda(O_1 = o_{k(1)} \cap S_1 = s_j) = \pi_j b_{jk(1)}$

・ $t \geq 2$ のとき $\alpha_t(j) = \bigcirc \alpha_{t-1}(i)$ の形にしたい。

$$\alpha_t(j) = \sum_{i(1)=1}^N \sum_{i(2)=1}^N \dots \sum_{i(t-1)=1}^N \pi_{i(1)} b_{i(1)k(1)} a_{i(1)i(2)} b_{i(2)k(2)} \dots a_{i(t-1)j} b_{jk(t)}$$

$i = i(t-1)$ とすると、

$$\begin{aligned} \alpha_t(j) &= \sum_{i(1)=1}^N \sum_{i(2)=1}^N \dots \sum_{i=1}^N \pi_{i(1)} b_{i(1)k(1)} a_{i(1)i(2)} b_{i(2)k(2)} \dots b_{ik(t-1)} a_{ij} b_{jk(t)} \\ &= \sum_{i=1}^N \left(\sum_{i(1)=1}^N \dots \sum_{i(t-2)=1}^N \pi_{i(1)} b_{i(1)k(1)} a_{i(1)i(2)} b_{i(2)k(2)} \dots b_{ik(t-1)} \right) a_{ij} b_{jk(t)} \\ &= \sum_{i=1}^N \alpha_{t-1}(i) a_{ij} b_{jk(t)} \end{aligned}$$

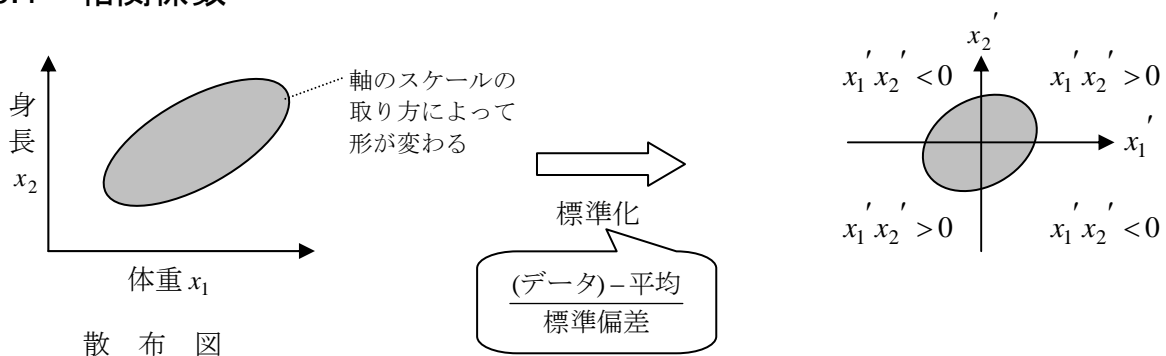
よって

$$P_\lambda(O_1 = o_{k(1)} \cap O_2 = o_{k(2)} \cap \dots \cap O_T = o_{k(T)}) = \sum_{i=1}^N \alpha_T(i)$$

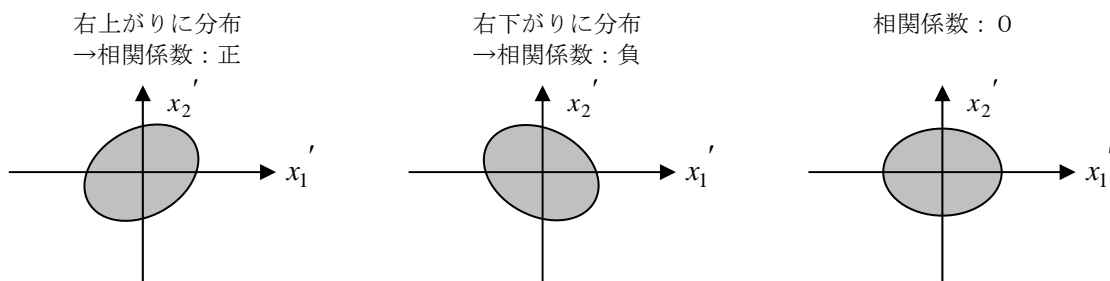
この方法は $N^2 T$ 程度の積和演算ですむ。

6 主成分分析

6.1 相関係数



相関係数 $r_{12} = x_1' x_2'$ の平均 $= \overline{x_1' x_2'} = \frac{1}{n} \sum_{i=1}^n x_1' x_2' = \frac{1}{n} \sum_{i=1}^n \frac{x_{1i} - \bar{x}_1}{\sigma_1} \frac{x_{2i} - \bar{x}_2}{\sigma_2}$



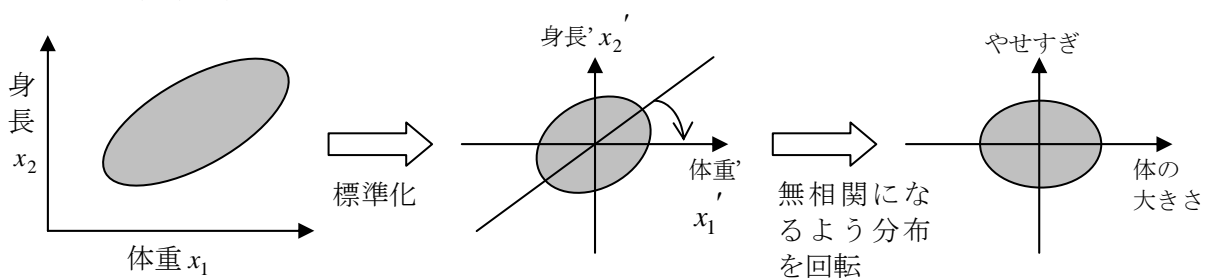
6.2 共分散行列

共分散 $a_{12} = \frac{1}{n} \sum_{i=1}^n (x_{1i} - \bar{x}_1)(x_{2i} - \bar{x}_2) = r_{12} \sigma_1 \sigma_2$

共分散行列 $A = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$ $a_{12} = a_{21}$

標準化されたデータ ($\sigma_1 = \sigma_2 = 1$) の共分散は相関係数と等しい。

6.3 主成分分析



r 個の観測データの組が n 個ある。これらのデータを主成分分析したい。

i 番目の観測データの組を $\vec{x}_i = \begin{bmatrix} x_{1i} \\ x_{2i} \\ \vdots \\ x_{ji} \\ \vdots \\ x_{ri} \end{bmatrix}$ とする。

(1) データの標準化

$$x'_{ji} = \frac{x_{ji} - \bar{x}_j}{\sigma_j}$$

(2) 共分散行列 $A=[a_{jk}]$ の計算

$$a_{jk} = \frac{1}{n} \sum_{i=1}^n x'_{ji} x'_{ki}$$

(3) 無相関になるように分布を回転

r 個の観測データの内、 j 番目 (次元目) と k 番目 (次元目) を任意に選び、回転行列 U を用いて $\bar{y}_i = U\bar{x}_i$ のように分布を回転させる。

回転後の共分散 $a'_{jk} = \frac{1}{n} \sum_{i=1}^n y_{ji} y_{ki}$ が無相関 $a'_{jk} = 0$ ($j \neq k$) になるように分布を回転させたい。

共分散行列 A の固有ベクトル $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_r$ を横に並べた行列を $\bar{X} = [\bar{X}_1 \bar{X}_2 \dots \bar{X}_r]$ としたとき、この回転行列 U は次式で得られる。

$$U = \bar{X}^{-1}$$

また、回転後の各軸方向の分散は、共分散行列 A の固有値 $\lambda_1, \lambda_2, \dots, \lambda_r$ となる。

【証明】

回転後の共分散行列は A' は次式のように変形できる。

$$\begin{aligned} A' = [a'_{jk}] &= \left[\frac{1}{n} \sum_{i=1}^n y_{ji} y_{ki} \right] = \frac{1}{n} \sum_{i=1}^n \bar{y}_i \cdot \bar{y}_i^T = \frac{1}{n} \sum_{i=1}^n (U\bar{x}) \cdot (U\bar{x})^T \\ &= \frac{1}{n} \sum_{i=1}^n U\bar{x}\bar{x}^T U^T = U \left(\frac{1}{n} \sum_{i=1}^n \bar{x}\bar{x}^T \right) U^T = UAU^T = UAU^{-1} \end{aligned} \quad (1)$$

ここで回転行列には $U^T = U^{-1}$ が成り立つことに注意する。この A' が $A' = \begin{bmatrix} a'_{11} & 0 & \dots & 0 \\ 0 & a'_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & a'_{rr} \end{bmatrix}$ のように、

対角以外の要素が 0 になるような回転行列 U を求めたい。

ところで、共分散行列 A の固有値を $\lambda_1, \lambda_2, \dots, \lambda_r$ 、固有ベクトル $\bar{X}_1, \bar{X}_2, \dots, \bar{X}_r$ を横に並べた行列を $\bar{X} = [\bar{X}_1 \bar{X}_2 \dots \bar{X}_r]$ としたとき、

$$A\bar{X} = A[\bar{X}_1 \bar{X}_2 \dots \bar{X}_r] = [A\bar{X}_1 A\bar{X}_2 \dots A\bar{X}_r] = [\lambda_1 \bar{X}_1 \lambda_2 \bar{X}_2 \dots \lambda_r \bar{X}_r] = \bar{X} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_r \end{bmatrix}$$

となる。両辺の左から \bar{X}^{-1} をかけると、

$$\bar{X}^{-1}A\bar{X} = \bar{X}^{-1}\bar{X} \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_r \end{bmatrix} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \lambda_r \end{bmatrix} \quad (2)$$

となる。(1)式と(2)式より、

$$U = \bar{X}^{-1}$$

$$a'_{ii} = \lambda_i \quad (\text{回転後の各軸方向の分散は共分散行列の固有値に対応})$$

が得られる。

参考文献

- [1] 上坂吉則, 尾関和彦 著, パターン認識と学習のアルゴリズム, 文一総合出版, 1990.
- [2] 奥村晴彦 著, パソコンによるデータ解析入門, 技術評論社, 1986.